APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: A PUBLISH-SUBSCRIBE SYSTEM

APPLICANT: RAINER RUGGABER, ELMAR DORNER, ALEXANDER
SCHMID AND KIM ELMS

A PUBLISH-SUBSCRIBE SYSTEM

TECHNICAL FIELD

The following description relates to network communications, and in particular to publish-subscribe systems.

BACKGROUND

5   A publish-subscribe system contains information producers and information consumers. Information producers publish events to the system. Information consumers subscribe to particular categories of events within the system. The publish-subscribe system provides delivery of published events to those information consumers that subscribes to the event. In the publish-subscribe system, a producer of the message does not know where the message will be received and a consumer of the message does not know from where the message was sent.

One type of publish-subscribe system is subject-based. In the subject-based publish-subscribe system, each message belongs to one of a fixed set of subjects. Producers label each message with a subject and consumers subscribe to messages having a particular subject.

Another publish-subscribe system is a content-based messaging (CBM) system. CBM systems support a number of information spaces, where subscribers may express a "query" against the content of published messages.

Publish-subscribe systems are unreliable because it is not known whether a message was actually received by a consumer or not. If the message is resent because some subscribers have not received the message, the remaining subscribers will receive a duplicate message.

SUMMARY

25   In one aspect, the invention is a publish-subscribe system. A publish-subscribe system includes a producer configured to publish a message and to subscribe to an

1

acknowledgement that the message was received. The system also includes a consumer configured to subscribe to the message and to publish the acknowledgement. The system further includes a network configured to register a message subscription and an acknowledgement subscription, to process the message subscription and the

5   acknowledgement subscription, to forward the message to the consumer based on the message subscription and to forward the acknowledgement to the producer based on the acknowledgement subscription.

This aspect may include one or more of the following features. The producer may include a reliability mechanism to subscribe to the acknowledgement associated with a

10   published message. The producer may include a timer and the producer may republish the message if a period of time expires before receiving the acknowledgement. The consumer may include a reliability mechanism to publish an acknowledgement for the message. The reliability mechanism may remove duplicate messages. The network may include a content-based messaging (CBM) router to route the message and the acknowledgement. The system

15   may use a generic addressing scheme in acknowledging a message. The generic addressing scheme may provides a unique identity to the consumer and the producer.

In another aspect, the invention is a reliability application program interface (API) for reliable content-based messaging. The API includes instructions to cause a processor to receive a message from a producer application, to register a subscription for an

20   acknowledgement to the message, to modify the message by adding fields to the message and to forward the message to an interface to publish the message to a content-based message (CBM) network. The instructions also cause a processor to receive the acknowledgement from the interface.

This aspect includes one or more of the following features. The API may include

25   instructions to cause a processor to implement a timer and republish the message if a period of time expires before receiving the acknowledgement. The fields may include an identification field and a message type field. The fields may include a recipient field that identifies one or more receivers of the message. The fields may include a sender field that identifies a sender of the message.

30   In another implementation, a reliability application program interface (API) for reliable content-based messaging is described. The API includes instructions to cause a

2

processor to receive a message from an interface, to modify the message by removing fields from the message, to publish an acknowledgement to the message, and to forward the message to a consumer application.

This aspect may include one or more of the following features. The API may include instructions to cause a processor to determine if the message has been previously received and to filter out the message if the message has previously been sent. The fields may include an identification field and a message type field. The fields may include a recipient field that identifies one or more receivers of the message. The fields may include a sender field that identifies a sender of the message.

One or more of the aspects above have one or more of the following advantages. The system has the advantage of being a reliable publish-subscribe system that ensures that messages are received. If the messages are not received, the system has the capability of resending the message only to those receivers that have not received the message. The system can be used with other systems that do not have a reliability mechanism.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of an exemplary communication system.

FIG. 2 is a block diagram of another exemplary communication system.

FIG. 3 is a process for delivering a message.

FIG. 4A is an example of a request message.

FIG. 4B is an example of an acknowledgement message.

FIG. 5 is a diagram of application layer interfaces.

FIG. 6 is an exemplary calling sequence.

FIG. 7 is graph for throughput of unicast messages using a reliable content-based messaging system.

FIG. 8 is a block diagram of a computer system on which the process of FIG. 3 may be implemented.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

Various conditions in a content-based messaging system (CBM) network may lead to message loss or interruption in communications. For example, the CBM network may lose messages or be unable to send messages if the messages are not processed in a timely manner

5 by a message receiver. Likewise, messages also may be lost due to network failure or other network conditions. In addition, messages in a typical CBM network are not addressed and sent to a specified receiver thereby a message receiver does not know the source of the received messages. As a result, applications in a typical CBM network are not able to determine whether a message was successfully delivered or received.

10 Many applications require reliable end-to-end communications to operate continuously and properly. For example, a message sender requires reliability in sending messages to message receivers, and message receivers require reliability in receiving sent messages. Reliable CBM, as described in detail below, provides reliable end-to-end communications while preserving the publish-subscribe paradigm and many of the benefits

15 associated with a CBM system.

System Overview

Referring to FIG. 1, a communications system 100 may include two or more client devices 110 (e.g., a client device 110a and a client device 110b) and a CBM network 120,

20 which includes at least one CBM router 125. The client devices 110a and 110b exchange data with the CBM network 120 using communications paths 140.

The client devices 110a and 110b may be operated by one or more users to access and exchange information with the CBM network 120. An example of client device 110a and 110b is a general-purpose computer capable of responding to and executing instructions in a

25 defined manner.

FIG. 1 shows one client device 110a as a producer and one client device 110b as a consumer; however, either client device 110a or 110b may function as an information producer, an information consumer, or both.

Any message that is published to the CBM network 120 is evaluated by the CBM

30 router 125 against any current registered subscriptions. If an incoming message conforms to

a given subscription, the message is forwarded to the associated consumer 110b by the CBM network 120.

Producer 110a and consumer 110b are anonymous with respect to each other. For example, a message may be routed as follows. Consumer 110b submits a subscription expressing what type of message content or information is desired. Consumer 110b may submit subscriptions to CBM network 120 at any time. Producer 110a publishes messages to CBM network 120 without providing any address or destination for the information. CBM-Router 125 compares any published message with registered consumer subscriptions to determine the consumers that should receive the published message. For the routing process, all elements of the published message are taken into consideration when comparing the published message with consumer subscriptions. The elements are, for example, name-value pairs and each message includes multiple name-value pairs. CBM-router 125 then forwards the message to any consumer 110b that the CBM router 125 determines is subscribed to the information in the message.

A reliability component, for example, a reliability layer (R-Layer), which includes an associated protocol, executed by the R-layer, is installed on each client device to provide reliable end-to-end communication within communications system 100.

The R-layer ensures reliability by providing an acknowledgement mechanism. The acknowledgement mechanism generates an acknowledgement for published messages when reliable communications are selected by a user or by an application. If an acknowledgement does not arrive within a timeout period, the message is resent.

The reliability mechanism is an extension to CBM. The R-layer enables either producer 110a or consumer 110b to request and provide reliable message delivery at any time. Whenever reliable communications are desired, the producer or consumer applications may activate the reliability mechanism to ensure reliable communication (e.g., through configuration, automatically, or through a user interface). However, the decoupling of the producer and consumer is maintained, because the producer does not know the consumers of a message.

Reliability Mechanism

To increase reliability, the R-layer provides a solution to various problems associated with CBM systems, for example, message loss and missed timeouts. Message loss occurs when a message is not delivered to a receiver/consumer because of various reasons, such a

5   network communications problems.

Message loss is solved by an acknowledgement mechanism. When reliable communications are desired, a timer may be associated with a published message. A message timeout (e.g., expiration of the timer) forces a producer 110a to resend a message. Therefore, if message delivery is not made within a given time, the producing application

10   receives an error message and the message is resent. The delivery of a message is considered reliable, if the producing application does not receive an error within a given time after the message is handed over to the R-layer using the send-method. As a result, the status of a message delivery is determined passively by monitoring for an error indication.

However, if messages are resent, consumer 110b can receive the same message more

15   than one time. For example, if a message times out too early, both the resent and the original message may be received by a consumer. Therefore, duplicate messages are identified by a message identification feature and are filtered out.

Architecture

The reliability protocol is provided by the client devices 110. As a result, no changes

20   to CBM network 120 are required to implement a reliable CBM system.

FIGS. 2 and 3 illustrate reliable delivery of a message. Producer 110a includes a producer application 210a, a R-Layer 215 and a CBM library 220a. Consumer 110b includes a consumer application 210b, a R-layer 225 and CBM library 220b. CBM library 220a interfaces with CBM network 120. For example, using CBM library 220a, producer 110a

25   may publish information to CBM network 120.

In order to provide reliable messaging for application 210b, R-Layer 215 and R-layer 225 each provides an interface between the application and CBM client library 220b. Each R-Layer 215 and 225 uses its respective CBM client library 220a and 220b for communication with CBM network 120.

Applications 210a and 210b are stored in their respective CBM library 220a and 220b. As a result, each application 210a and 220b can communicate reliably (using the R-Layer interface in conjunction with the CBM library). Applications 210a and 210b may also interface directly and unreliably with CBM library 220a and 220b respectively by bypassing

5   R-Layers 215 and 225.

R-Layers 215 and 225 each use their respective CBM client library 220a and 220b as an unreliable messaging service. In the following description, the functionality of the CBM client library 220 is bypassed so that each R-Layer 215 and 225 is communicating directly with CBM-Router 125.

10   Referring to FIG. 3, process 300 is an exemplary process for delivering a reliable message using communications system 100.

Process 300 provides subscriptions to CBM network 120 from consumer 110b (302). The subscription expresses what type of message content is of interest to the consumer application 210b. When reliable communication is desired, process 300 extends (304) the

15   subscription. For example, the subscription is extended by R-layer 225 without restricting the original subscription. Process 300 submits (306) the subscription by using R-Layer 225 on behalf of consumer application 210b to CBM network 120 using CBM library 220b.

Process 300 submits a subscription for acknowledgments to the CBM network 120 by using R-Layer 215 through its CBM library 220a (308). The producer R-layer 215

20   subscribes to acknowledgements to messages published by the producer application 210a.

When producer 110a publishes a message for which reliability is desired, process 300 provides the message to R-Layer 215 (310). Process 300 adds one or more fields by using the reliability mechanism of the R-layer 215 as required by the R-layer to modify the message(312). The fields are used by an addressing scheme of the acknowledgement

25   mechanism and a message identification that allows specific messages to be identified in the communications network 100. The modified message conforms to CBM standards (i.e., every field within the message can be examined for routing of the message).

Process 300 provides the modified message to the CBM library 220a using R-layer 215 (314). Process 300 sends the message from CBM library 220a to the CBM-router 125

30   through communications path 140 (316). Process 300 submits a subscription from the

producer R-layer 215 to the CBM router 125 to register for acknowledgement messages in response to the published messages for which reliability is desired (318).

Process 300 compares the published message with all consumer subscriptions to determine the consumers of the message (320). For example, the comparison is performed in the CBM-router 125. Process 300 forwards the published message to client devices 110 according to the subscriptions determined by CBM-router 125 (322).

Process 300 receives messages at the consumer R-layer via CBM library 220b to which the consumer application 210b has subscribed (324). Process 300 publishes an acknowledgement message for each received message from the consumer R-layer 225 (326). Process 300 truncates the added reliability fields using the consumer R-layer 225 (328). Process 300 forwards the original message (i.e., the unmodified message) to the consumer application (330).

Process 300 sends the published acknowledgement message to the CBM router 125 (332). Process 300 determines the R-layer (associated with a producer) that subscribes to the acknowledgement message (334). For example, the determination is made by the CBM router 125. Process 300 sends the acknowledgement message from CBM router 125 to the corresponding producer R-Layer 215 (i.e., the associated producer application) to ensure the reliable delivery of the message (336).

The producer R-layer 215 implements an acknowledgement receipt timer. If the acknowledgement messages is not received, a timeout period expires and the R-layer 215 may resend the message to ensure reliable delivery of the message.

Protocol Fields and Subscriptions

In order to execute the protocol used by the R-layer, the messages are supplemented with additional information and subscriptions are added to ensure the required messages are received. New name-value pairs are required to be able to describe required messages using subscriptions. In the following description, the added name-value pairs will be provided before the subscriptions are presented.

Referring to FIGS. 4A and 4B, a request message 410 and an acknowledgement message 430 each includes a R-Consumer field 412, a R-Producer field 414, a R-MID field 416 and a R-type field 418.

The R-Consumer field 412 identifies those one or more receivers from whom the sender of the message expects an acknowledgement. Multiple receiver names are separated by a "|"-symbol. This enables the unique identification of receivers in the string. The "|"-symbol is reserved and cannot be used in a receivers identification.

5 The R-Producer field 414 contains the identification of the sender of the reliable message. No separation symbol is needed, as there can be only one sender for a message.

The R-MID field 416 is a message identification field. The R-MID field 416 contains a sequence number that uniquely identifies a message sent by a specific sender. The receiver of a reliably sent message stores the R-MID – sender pairs for all messages it has received.

10 If a message is received with an R-MID – sender pair that is already in the list, then the message is considered a duplicate and discarded.

The R-Type field 416 identifies a message to be a request or an acknowledgement. A value of "ACK" indicates a request message and a value of "REQ" indicates a request message.

15 The specified name-value pairs are needed by each R-Layer 215 and 225 for the sender and the receiver of the messages to be able to specify the required messages.

The Producer 110a uses the subscription:

"(R-Producer == 'ProducerX') && R-Type == 'ACK'"

to receive all acknowledgements from messages sent by the producer 110a. The Producer

20 110a requires that the R-Producer field 414 contains its identification and that the type of the message, contained in the R-Type field 418 is set to "ACK".

The Consumer 110b uses the subscription:

"(Application Subscription) && contains(R-Consumer,'|ConsumerX|') && R-Type == 'REQ'")

25

to specify all messages it wants to receive. The receiver of a reliably sent message requires messages to fulfill the original subscription of the receiving application, to have the message type (R-Type) 418 be "REQ", and to have the R-Consumer field 412 contain the identification of the receiver.

9

Generic addressing scheme

The generic addressing scheme is an extension of the traditional use of addresses identifying communication endpoints. Addressing identifies components that provide acknowledgement messages (ACKs). In the R-Layers 215 and 225, it is a mutual agreement

5   that creates a context between loosely coupled senders and receivers. The R-Layers 215 and 225 do not define the structure of the addresses. Addresses can be provided by applications using addressing schemes that are already in existence.

To ensure reliability the producer R-Layer 215 determines each consumer R-Layer 225 from which an acknowledgement is expected. Likewise, the consumer R-Layer 225 has

10   to determine for which messages an acknowledgement is to be sent. To provide for these determinations a generic addressing scheme is introduced to identify communication endpoints. The addressing scheme provides each client or application with a unique identity. The addressing scheme is generic; therefore, no mapping of existing address formats is required and existing addressing schemes can be used directly.

15   In addition, the addressing scheme is implemented in the R-Layers 215 and 225 to realize the acknowledgement mechanism, and is independent of the producer and consumer applications 210a and 210b. The producer and consumer applications 210a and 210b do not require any interface or special instructions when implementing the addressing scheme in the R-layers 215 and 225.

20   The generic addressing scheme is a string that can contain the address chosen by the application, for example, a number, an alphanumeric number, an Internet address and so forth.

Application Program Interface

25   Each R-Layer 215 and 225 has an application program interface (API), which is similar to the API that is provided by the underlying CBM library layer 220a and 220b thereby allowing transparent integration of their respective R-layer 215 and 225 into the communications system 100. As a result, integrations of existing applications are possible with minimal effort. The flexibility of the R-Layers 215 and 225 addressing scheme allows

30   the use of existing application addresses that are used in order to identify the R-Layers instances.

Referring to FIG. 5, producer 110a includes a R-layer Service Access Point (R-SAP) 510a and a CBM-SAP 520a and consumer 110b includes a R-SAP 510b and a CBM-SAP 520b.

The R-SAPs 510a and 510b are positioned between their respective Application layer 210a and 210b and their respective R-layer 215 and 225. At the producer application 210a, a command r_notify() publishes a message and specifies receiving R-Consumers and the command r_error() provides an error indication if the message cannot be sent within a given time. At the consumer application 210b, the command r_arrive() provides a message arrival event to the consumer application.

The CBM-SAPs 520a and 520b provide an interface between the R-layer 215 and 225 and the CBM layer. At the producer 110a, the command c_notify() publishes the message to the CBM network 120. At the consumer 110b, the command c_arv() provides for a message arrival event.

Referring to FIG. 6, a process 600 are the commands executed at the producer 110a and the consumer 110b to provide reliable messaging.

At the producer 110a, command r_producer() 602 creates the R-Producer field 414 and monitors for R-Layer error events. A subscribe() command 604 provides a subscription to CBM network 120 for acknowledgements. A r_notify() 606 sends the message and specifies the R-Layer consumers 110b expecting an acknowledgement.

At the consumer 110b, a command r_consumer() 612 creates the R-Consumer field 412. The command arv_action() 614 then specifies a receiver for arriving messages to handle the data in the consumer application 210b. A subscribe() command 616 provides a subscription to specific message content that is of interest to the consumer application 210b. The consumer application 210b then waits for published messages.

Multicast Messages

In the multicast case, reliable communication with more than one consumer is possible. One message is published and all specified consumers send an acknowledgement. If all acknowledgements do not arrive within a given time, the message is resent. To minimize overhead associated with resending messages, only consumers with outstanding

acknowledgements are taken into account. The unicast case can be seen as a specialization of the multicast case, because only one consumer is sending an acknowledgement.

The timeout mechanism is dynamic and adapts to varying network conditions. For the timeout calculation, recent message roundtrip times may be taken into consideration.

5 Therefore, the actual timeout variable is the moving average of recently successful delivered messages. If a timeout is missed, then the timeout calculation increases by a multiplicative fault term.

Each R-Layer 215 and 225 is continuously measuring the roundtrip time t. This is the time required to send a message to a remote site and return it again. The timeout for a reliable

10 message is 2*t. If the sender does not receive an acknowledgement after this timeout, the message is resent and the timeout is set to 4*t. The timeout is doubled every time it is missed until a specifiable upper limit. The upper limit ensures that there is a point in time when the sender can assume that the message was received.

Scalability with a growing number of clients is achieved by keeping the CBM

15 infrastructure simple. As previously described, the protocol logic is located at the client side, which provides adequate processing power to scale for a growing number of clients.

Flow control is realized by a send window mechanism, where only a specific amount of unacknowledged messages is allowed. Congestion is avoided through status messages from the CBM-Infrastructure. If the producer or consumer receives a dropped message

20 warning, indicating that the CBM infrastructure is overloaded, the producer can reduce the send rate (thereby minimizing the chance of congestion).

Heterogeneous environments

Support for heterogeneous environments is available, and applications with or without reliability requirements may be supported. Moreover, applications without reliability

25 requirements can receive messages from producers with reliability requirements; however, the message delivery is not reliable (i.e., there is no way for the producer to verify the message has been delivered). Each R-layer 215 and 225 extends messages by the fields added to the message, however, the modified messages conform to CBM standards and can be routed by regarding all message elements. In heterogeneous environments, messages that

30 are resent could harm consumers that are not implementing the R-Layer. In fact, the

12

consumers would receive the same massage multiple times. As a result, this would be a breach of the once delivery requirement provided by most CBM infrastructures. To prohibit multiple deliveries of the same message, messages are sent in a resend tunnel encapsulating the original messages.

5   In one implementation, the CBM infrastructure used for the implementation is Elvin. Elvin is a flexible and simple to use messaging architecture developed by the DSTC in Brisbane (Australia). The protocol works asynchronously (i.e., the producer loses the message context immediately after calling the send method); in particular the send method does not wait until the acknowledgement arrives.

10   The reliable multicast message delivery is only as fast as the slowest consumer. The consumer with the largest response time dictates the system performance.

        FIG. 7 shows that the reliability CBM protocol may adapt to changing conditions. Small unicast messages are sent over a system 100 consisting of standard hardware, e.g., producer 110a, consumer 110b and CBM router 125 run on Pentium II machines (550-

15   933MHz) connected by a 100MBit/s Ethernet. In this example, a throughput of 2300 messages per second is achieved. The example shows that the timeout adapts to varying conditions, and that the maximum queue length for unacknowledged messages was 1000.

        When message throughput decreases and the number of timeouts increase, system 100 adapts to reduce message timeout until throughput increases while maintaining queue

20   length (e.g., between 3886 and 6259ms). In the multicast case, the scalability for up to 1000 consumers was confirmed. With an increasing number of consumers, the total throughput of messages, consisting of request messages and acknowledgements was constant.

        Consumer subscriptions are preserved. The reliability mechanism extends the subscriptions, without any harm to the original subscriptions. Consumers can express their

25   demands with the full functionality of the subscription language.

        The communications system 100 maintains loosely coupled consumer and producer relationship and the unrestricted use of client subscriptions is ensured. The communications system 100 also provides a connectionless communication with at least one delivery semantic. The communications system 100 also supports multicast communication, and runs

30   in heterogeneous environments, in which either client requires reliable and unreliable communication. The CBM system provides support for transparent integration of existing

applications. The CBM network 120 also provides an API that is similar to existing CBM APIs requiring little if any additional programming overhead to allow integration in existing systems.

The CBM system 100 also adapts to varying network conditions and client

5   capabilities. In addition, the CBM system may be scaled to the number of consumers and message size.

Problems due to message loss are transparent to the client devices 110 and applications 210. Consumer application 210b receives messages reliably through a subscription for message content, exactly in the same way as the unreliable message. The

10   reliability mechanism masks all faults and provides a reliable messaging service for the applications of the client devices 110.

FIG. 8 shows a computer 800 for using process 300. Computer 800 includes a processor 802, a volatile memory 804, and a non-volatile memory 806 (e.g., hard disk). Non-volatile memory 806 stores operating system 810, data storage 812, and computer

15   instructions 814 which are executed by processor 802 out of volatile memory 804 to perform process 300.

Process 300 is not limited to use with the hardware and software of FIG. 8; it may find applicability in any computing or processing environment and with any type of machine that is capable of running a computer program. Process 300 may be implemented in

20   hardware, software, or a combination of the two. For example, process 300 may be implemented in a circuit that includes one or a combination of a processor, a memory, programmable logic and logic gates. Process 300 may be implemented in computer programs executed on programmable computers/machines that each includes a processor, a storage medium or other article of manufacture that is readable by the processor including

25   volatile and non-volatile memory and/or storage elements), at least one input device, and one or more output devices. Program code may be applied to data entered using an input device to perform process 300 and to generate output information.

Each such program may be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. However, the programs

30   can be implemented in assembly or machine language. The language may be a compiled or an interpreted language. Each computer program may be stored on a storage medium or

device e.g., CD-ROM, hard disk, or magnetic diskette that is readable by a general or special purpose programmable computer for configuring and operating the computer when the storage medium or device is read by the computer to perform process 300. Process 300 may also be implemented as one or more machine-readable storage media, configured with a

5 computer program(s), where upon execution, instructions in the computer program(s cause a computer to operate in accordance with process 300.

Process 300 is not limited to the specific implementations described herein. For example, process 300 is not limited to the specific processing order of FIG. 3. Rather, the blocks of FIG. 3 may be re-ordered, as necessary, to achieve the results set forth above.

10 In some implementations, client devices 110 also may include a special-purpose computer, a personal computer ("PC"), a workstation, a server, a laptop, a Web-enabled phone, a Web-enabled personal digital assistant ("PDA"), an interactive television set, a set top box, an on-board (i.e., vehicle-mounted) computer, or a combination of two or more these devices capable of responding to, generating, and/or executing instructions. The client

15 device 110 may include any number of other devices, components, and/or peripherals, such as memory/storage devices, input devices, output devices, user interfaces, and/or communications interfaces.

The client device 110 also may include one or more software applications (e.g., an operating system, a browser application, a microbrowser application, a server application, a

20 proxy application, a gateway application, a tunneling application, and a CBM communications application) loaded on the client device 110 to command and direct the client device 110. Applications may include a computer program, a piece of code, an instruction, or some combination thereof, for independently or collectively instructing the client device 110 to interact and operate as desired.

25 The applications may be embodied permanently or temporarily in any type of machine, component, physical or virtual equipment, storage medium, or propagated signal capable of providing instructions to the client 110 device. In particular, the applications may be stored on a storage medium or device (e.g., a read only memory (ROM), a random access memory (RAM), a volatile/non-volatile memory, a magnetic disk, or a propagated signal)

30 readable by the client device 110, such that if the storage medium or device is read by the client device 110, the specified step or instructions are performed.

The client device 110 also may include one or more communications interfaces that allow the client device to send and receive information using the communications paths 140.

The communications paths 140 may be configured to send and receive signals (e.g., electrical, electromagnetic, or optical) that convey or carry data streams representing various types of analog and/or digital content. For example, the communications paths 140 may be implemented using various communications media and one or more networks comprising one or more network devices (e.g., servers, routers, switches, hubs, repeaters, and storage devices). The one or more networks may include a WAN, a LAN, a plain old telephone service (POTS) network, a digital subscriber line (DSL) network, an integrated services digital network (ISDN), and a synchronous optical network (SONET), or a combination of two or more of these networks. In addition, the communications paths 140 may include one or more wireless links that transmit and receive electromagnetic signals, such as, for example, radio, infrared, and microwave signals to convey information.

The communications system 100 uses a publish-subscribe paradigm or technique to exchange data between the client devices 110. According to this technique, each client device 110 may be an information consumer or producer. As an information producer, the client device 110 may publish information to the CBM network 120. As an information consumer, the client device 110 registers a subscription to specific information published to the CBM network 120. The CBM network 120 provides notifications of the published information (e.g., as messages) to those consumer client devices 110 that have registered subscriptions to the information.

In some implementations, the CBM network 120 may be implemented using a publish-subscribe system, such as, for example, Elvin, CosNotif, JMS, Keryx, and Gryphon to distribute information to the client devices 110 and their associated communications applications. In one implementation of the communications system 100, the Elvin CBM infrastructure is used.

In some implementations, CBM network 120 may include one or more data processing and distribution devices (e.g., a server, a router, and associated communications media and data transport systems). For example, CBM network 120 may include one or more CBM routers 125 and communications interfaces (not shown) that receive the

published information and generate messages that are transmitted to the consumers who subscribe to the information.

CBM router 125 may be implemented by a processor and a CBM routing application that exchanges information with the communications interfaces (which are configured to communicate with the communications paths 140).

Other implementations are within the scope of the following claims.